Assignment 07

(5 points)

In this assignment, you use the insurance data set (downloadable from HuskyCT).

You implement a support vector machine classifier.

In the code you will see that the target variable y is converted to have values of -1 and 1. Numeric variables are selected. The variables are normalized to make sure the distance-based measures would not be distorted by the range differences between variables. 60% observations are used for training the SVM classifier, the rest is used to test it. Stratified sampling is used. Training data are stored in XTrain and yTrain. (XTrain: predictor variables, yTrain: target variable).

The training data is first projected on a selected set of eigenvectors (which retains the total variance of beta percent), this is implemented in pca function. The returned XTrain has a smaller number of dimensions (or columns). XTest is projected using the w transform (formed by the selected eigenvectors).\

Gaussian kernel is used.

The primal QP optimization problem for a non-separable case

$$\min_{w} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \theta^{(i)}$$

Subject to

$$y^{(i)}\left(w^T x^{(i)} + b\right) - 1 + \theta^{(i)} \geq 0 \quad \text{and } \theta_i \geq 0 \ \forall \text{i}$$

And the corresponding dual optimization problem

$$\max_{\alpha} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)^T} x^{(j)} \right] =$$

Subject to

$$C \geq \alpha_i \geq 0, \ i = 1 \ldots N$$

$$\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

C is set to a large number to penalize more to the points falling between the margins.

In the vectorized form:

$$\underset{\alpha}{arg\max}[\mathbf{1}^T * \boldsymbol{\alpha} - 0.5\boldsymbol{\alpha}^T * diag(\boldsymbol{y}) * \boldsymbol{X} * \boldsymbol{X}^T * diag(\boldsymbol{y}) * \boldsymbol{\alpha}]$$

Subject to

$$\boldsymbol{I\alpha} \geq \mathbf{0}$$

$$\boldsymbol{\alpha}^T \boldsymbol{y} = 0$$

In which **1** is a column vector of all values 1, $\boldsymbol{\alpha}$ is a column vector of α_i, **y** is the column vector of the target variable in the training data set, and **X** is the training data set with all the variables on its columns and observations on its rows. **I** is the identity matrix.

The term

$$\boldsymbol{X} * \boldsymbol{X}^T$$

Can be denoted **K**, which is the linear kernel, so linear kernel also means no kernel. The above becomes

$$\underset{\alpha}{arg\max}[\mathbf{1}^T * \boldsymbol{\alpha} - 0.5\boldsymbol{\alpha}^T * diag(\boldsymbol{y}) * \boldsymbol{K} * diag(\boldsymbol{y}) * \boldsymbol{\alpha}]$$

Subject to

$$\boldsymbol{I\alpha} \geq \mathbf{0}$$

$$\boldsymbol{\alpha}^T \boldsymbol{y} = 0$$

Other kernels include a polynomial kernel

$$\mathrm{K}\left(x^{(i)}, x^{(j)}\right) = \left({x^{(i)}}^T x^{(j)} + 1\right)^d$$

Or a Gaussian kernel

$$\mathrm{K}\left(x^{(i)}, x^{(j)}\right) = exp\left(\frac{-\|x - x_k\|^2}{2\sigma^2}\right)$$

After getting $\boldsymbol{\alpha}$ from the optimization function, only $\alpha_i > 0$ is kept. $x^{(i)}$ corresponding to nonzero α_i are called support vectors.

A new observation x can be classified as follows:

$$g(x) = \sum_{j=1}^{N} \alpha_j y_j K\left(x, x^{(j)}\right) + b$$

where

$$b = \frac{1}{N_s}\left(\sum_{i=1}^{N_s}\left(y_i - \sum_{j=1}^{N_s} \alpha_i y_i K\left(x^{(i)}, x^{(j)}\right)\right)\right)$$

This is because any support vector satisfies $y_i g(x_i)=1$, so

$$y_i\left(\sum_{j=1}^{N_s} \alpha_j y_j K\left(x^{(i)}, x^{(j)}\right) + b\right) = 1$$

Knowing that $y_i^2=1$, so from there we get b

1. (4 points) Read, understand, and run the MATLAB code.

Pick either R or Python to do the work based on the working MATLAB code is given to you. 1 additional point if you write code in both Python, and R. Logics, variable names, and function names should follow the ones in the given MATLAB code as much as possible.

2. (1 point) Explain your code (you can do this line by line, or small blocks of code).

Note: The negative sign in f in the code as quadprod is a minimization problem while we are maximizing the objective function. Also note that you can comment out matlabpool stuff if you do not have or do not want to run code on multiple cores on your computer.

Submission: your script, and a document explaining your work.